

# Industrielle Bildverarbeitung

## 1.Übung Sommersemester 2012

Pham Hai Dang Le und Michael Grünwald

HTWG Konstanz, Brauneggerstr. 55, D-78462 Konstanz, Deutschland

<http://www.ios.htwg-konstanz.de/>

# Overview

1 Ablauf

2 Matlab

3 MEX

4 Übung

# Overview

1 Ablauf

2 Matlab

3 MEX

4 Übung

# Übungsablauf

- Zeit:

15:30 Uhr bis 17:00 Uhr

- Raum:

F 112 und F 114

- Email:

dangle@htwg-konstanz.de

- Download

Laborübungen

# Overview

1 Ablauf

2 Matlab

3 MEX

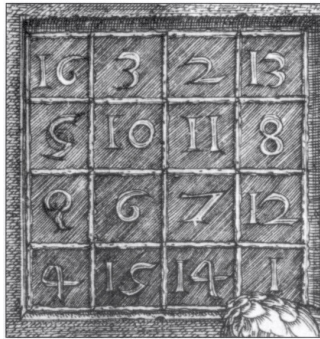
4 Übung

## 2.1 Matlab: Einführung

- Matlab steht für *MATrix LABoratory*.
- Basiselement ist ein Array.
- Matlab unterstützt Berechnungen von Matrizen und Vektoren.
- Matlab ist eine riesige Sammlung an numerischen und grafischen Funktionen.
- Funktionen sind selber in der Matlab-Skript-Sprache oder in C oder Fortran geschrieben.

## 2.2 Matlab: Einführung

- Folgende Beispiele sind aus MATLAB-"Getting Started Guide"
- Albrecht Dürer (21.Mai 1471 – 6.April 1528), deutscher Maler, Grafiker, Mathematiker und Kunsttheoretiker
- Melencolia I, Kupferstich (1514)
- Magisches Quadrat



## 2.3 Matlab: Matrix

- Dürers Matrix erstellen:

$$A = [16 \ 3 \ 2 \ 13; 5 \ 10 \ 11 \ 8; 9 \ 6 \ 7 \ 12; 4 \ 15 \ 14 \ 1]$$

- `sum(A)`
- `A'`
- `sum(A')`
- `sum(A')'`
- `diag(A)`
- `sum(diag(A))`
- Nacheinander eingeben: `A(1) A(2) A(3)`
- `A(1,4) + A(2,4) + A(3,4) + A(4,4)`
- `A(1:4)` und `sum(A(1:4))`
- `A = magic(4)`



## 2.4 Matlab: Funktionen in Matlab

- Kreieren Sie die Datei `addiere.m`
- Definieren Sie die Anzahl von Eingabe- und Ausgabeparametern:  
`[ergebnis] = addiere(zahl1, zahl2)`
- Implementieren und testen Sie die Funktion `addiere.m`
- Erstellen Sie Kommentare: `%`
- Bilder einlesen: `image = imread(FILENAME)`
- Analysieren Sie die Funktion: `show.m`
- `help [command]`
- `help show.m`

## 2.5 Matlab: Erweiterung

- Wofür benötigt man die Funktionen *tic* und *toc*?
- Nennen Sie Vorteile und Nachteile von Matlab?
- help mex
- MEX     Compile MEX-function

Usage:

mex [options ...] file [files ...]

Description: ...

# Overview

1 Ablauf

2 Matlab

**3 MEX**

4 Übung

## 3.1 MEX: Einführung

- mex -setup
- Pick:
  - Template Options file for building gcc MEXfiles (Linux)
  - MSVC / Visual C Compiler (Windows)
- MEX Dateien können von Matlab aus aufgerufen werden und besitzen daher eine ähnliche Struktur wie Matlab Dateien.

## 3.2 MEX: Grundstruktur

```
#include "mex.h"

/*=====
 * Entry point to C MEX-file
 *=====
 * prhs: An array of right-hand input arguments
 * plhs: An array of left-hand output arguments
 * nrhs: The number of right-hand arguments, or the size of the prhs array
 * nlhs: The number of left-hand arguments, or the size of the plhs array
 *=====*/
void mexFunction( int nlhs, mxArray *plhs[],
                  int nrhs, const mxArray *prhs[])
{
    ...
}
```

- Übertragen Sie die Eingabe- und Ausgabeparameter von der addiere.m Datei auf die MEX Datei.



## 3.3 MEX: Deklaration

```
int dims[2];  
int height;  
int width;  
int x;  
int y;  
unsigned char* pSource;  
unsigned char* pDest;
```

## 3.4 MEX: Überprüfung der Parameter

```
// check parameters, only allow 1
if(nlhs > 1 || nrhs != 1)
    mexErrMsgTxt("Usage: imageOut = mextemplate(imageIn)");

// Get image size, don't mess up dims 1 and 0!
height = dims[1] = mxGetN(prhs[0]);
width = dims[0] = mxGetM(prhs[0]);
```

## 3.5 MEX: Überprüfung der Parameter

```
// we only support 8 bit images (char*)
if(mxIsUint8(prhs[0]))
    mexErrMsgTxt („Entrypoint can only process UINT8.");

// output
plhs[0] = mxCreateNumericArray(2, dims, mxUINT8_CLASS, mxREAL);

//input
pSourceBase = (unsigned char*) mxGetData(prhs[0]);
pDestBase = (unsigned char*) mxGetData(plhs[0]);
// To Do: copy possibly modified image data to right hand side
results (output)
```



## 3.6 MEX: Ausführung (Execute)

- mex **function\_name**.cpp
- **image** = imread('name.jpg');
- imagesc(**function\_name**(**image**));

# Overview

1 Ablauf

2 Matlab

3 MEX

4 Übung

# Übung

- Stellen Sie das Bild auf dem Kopf!
- Implementieren Sie die fehlenden Funktionen.
- Die Übungen dürfen zu zweit bearbeitet werden.
- Jeder muss seine Lösung vorstellen können.
- Lösungen werden begutachtet, wenn die folgenden Bedingungen erfüllt sind:
  - Alle Aufgaben werden am Dienstag überprüft.
  - Für die Identifikation müssen Übungsaufgabennummer, Nachname, Vorname, Matrikelnummer, Semester, Email-Adresse und Datum von allen beteiligten Studenten angegeben werden.
  - Die Lösungen dürfen nur eingereicht werden, wenn Sie die Lösung selbst und selbständig bearbeitet haben.
  - Das Einreichen fremder Lösungen (Lösungen von Dritten, aus Büchern, dem Internet oder anderen Quellen) ist ein Täuschungsversuch und wird als nicht bestanden bewertet.